



永



The fact that tuples belong to sequence types means:

they are actually lists

they can be extended using the .append() method

they can be modified using the del instruction

they can be indexed and sliced like lists





市 流

Item 2/20

What is the output of the following snippet?

def fun(x): x += 1return x x = 2x = fun(x + 1)print(x)



5

4

the code is erroneous



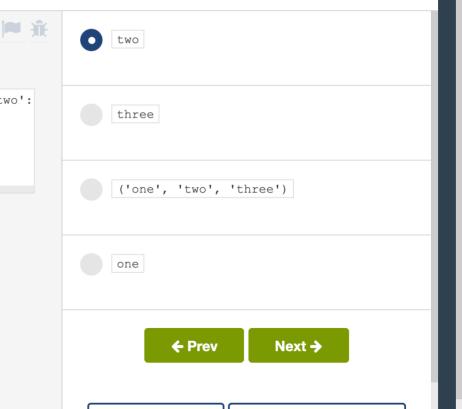




≡ Item 3/20

What is the output of the following snippet?

```
dct = { 'one':'two', 'three':'one', 'two':
    v = dct['one']
for k in range(len(dct)):
       v = dct[v]
print(v)
```







i



Assuming that tuple is a correctly created tuple, the fact that tuples are immutable means that the following instruction:

tuple[1] = tuple[1] + tuple[0]



is illegal



can be executed if and only if the tuple contains at least two elements



is fully correct



may be illegal if the tuple contains strings







₩ 第

≡ Item 5/20

The following snippet:

def func(a, b):
 return a ** a

print(func(2))

- will output 4
- is erroneous
- will output 2
 - will return None







市 流

≡ Item 6/20

What is the output of the following snippet?

def fun(inp=2, out=3):
 return inp * out
print(fun(out=2))











the snippet is erroneous









市

Item 7/20

What is the output of the following snippet?

def f(x):if x == 0: return 0 return x + f(x - 1)print(f(3))



1

the code is erroneous







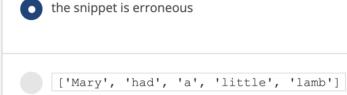


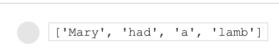
₩ 第

≡ Item 8/20

What is the output of the following snippet?

```
list = ['Mary', 'had', 'a', 'little', 'la
def list(lst):
    del lst[3]
    lst[3] = 'ram'
print(list(list))
```













₩ 第

≡ Item 9/20

The following snippet:

def func1(a):
 return a ** a

def func2(a):
 return func1(a) * func1(a)

print(func2(2))

is erroneous

will output 4

will output 2

will output 16

← Prev





₩ 第

Item 10/20

What is the output of the following snippet?

def fun(x): if x % 2 == 0: return 1 else: return print(fun(fun(2)) + 1)





2



1

the code will cause a runtime error







≡ Item 11/20

A function defined in the following way:

def function(x=0):
 return x

- must be invoked without arguments
- may be invoked with any number of arguments (including zero)
- may be invoked without any argument, or with just one

must be invoked with exactly one argument

← Prev



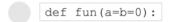


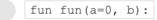
₩ 第



Which of the following lines properly starts a function using two parameters, both with zeroed default values?















≡ Item 13/20

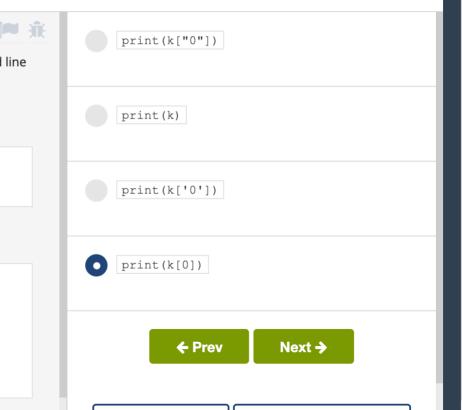
What code would you insert into the commented line to obtain the output that reads:

Expected output:

a b c

Code:

dct = { }
lst = ['a', 'b', 'c', 'd']
for i in range(len(lst) - 1):
 dct[lst[i]] = (lst[i],)
for i in sorted(dct.keys()):
 k = dct[i]
 # insert your code









A built-in function is a function which:

- has been placed within your code by another programmer
 - has to be imported before use
- comes with Python, and is an integral part of Python
- is hidden from programmers







₩ 第

Item 15/20

What is the output of the following snippet?

tup = (1, 2, 4, 8)tup = tup[1:-1]tup = tup[0]print(tup)

















市 流

Item 16/20

What is the output of the following snippet?

def any(): print(var + 1, end='') var = 1any() print(var)























₩ 第

≡ Item 17/20

What is the output of the following snippet?

def fun(x):
 global y
 y = x * x
 return y

fun(2)
print(y)





None



the code will cause a runtime error



2









I

≡ Item 18/20

Which of the following statements is false?





The None value cannot be used as an argument of arithmetic operators

The None value may not be used outside functions







₩ 第

Item 19/20

What is the output of the following snippet?

def fun(x, y, z): return x + 2 * y + 3 * zprint(fun(0, z=1, y=3))















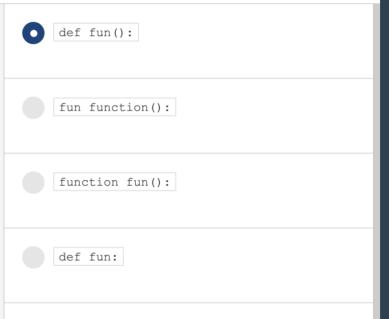




永

≡ Item 20/20

Which of the following lines properly starts a parameterless function definition?



← Prev